
Polygon API Access

Julius Dsouza

Nov 18, 2022

PYTHON API:

1	Polygon API Access	1
2	Indices and tables	3
	Index	5

POLYGON API ACCESS

class `polygon_api_access.PolygonAPIAccess(location, table_name)`

Instantiate the PolygonAPIAccess class and takes in 2 parameters.

Parameters

- **location** (*str*) – The file path to store the database
- **table_name** (*str*) – Name of the database

access(*currency_pairs*)

Parameters

currency_pairs (*list*) – A dictionary defining the set of currency pairs we will be pulling data for

aggregate_raw_data_tables(*engine, currency_pairs*)

This function is called every 6 minutes to aggregate the data, store it in the aggregate table, and then delete the raw data

Parameters

- **engine** (*sqlalchemy.future.engine.Engine*) – Engine to connect to the database
- **currency_pairs** (*list*) – A dictionary defining the set of currency pairs we will be pulling data for

initialize_aggregated_tables(*engine, currency_pairs*)

This creates a table for storing the (6 min interval) aggregated price data for each currency pair in the SQLite database

Parameters

- **engine** (*sqlalchemy.future.engine.Engine*) – Engine to connect to the database
- **currency_pairs** (*list*) – A dictionary defining the set of currency pairs we will be pulling data for

initialize_raw_data_tables(*engine, currency_pairs*)

This creates a table for storing the raw, unaggregated price data for each currency pair in the SQLite database

Parameters

- **engine** (*sqlalchemy.future.engine.Engine*) – Engine to connect to the database
- **currency_pairs** (*list*) – A dictionary defining the set of currency pairs we will be pulling data for

reset_raw_data_tables(*engine, currency_pairs*)

Function which clears the raw data tables once we have aggregated the data in a 6 minute interval

Parameters

- **engine** (*sqlalchemy.future.engine.Engine*) – Engine to connect to the database
- **currency_pairs** (*list*) – A dictionary defining the set of currency pairs we will be pulling data for

reset_raw_data_tables2(*engine, currency_pairs*)

send_response(*from_, to, amount, precision*)

Open a RESTClient for making the api calls

ts_to_datetime(*ts*) → str

Function slightly modified from polygon sample code to format the date string

Parameters

ts (*datetime.date*) – Time Stamp

Returns

Formatted Time Stamp String

Return type

str

INDICES AND TABLES

- `genindex`
- `modindex`
- `search`

INDEX

A

`access()` (*polygon_api_access.PolygonAPIAccess*
method), 1

`aggregate_raw_data_tables()` (*poly-*
gon_api_access.PolygonAPIAccess *method*),
1

I

`initialize_aggregated_tables()` (*poly-*
gon_api_access.PolygonAPIAccess *method*),
1

`initialize_raw_data_tables()` (*poly-*
gon_api_access.PolygonAPIAccess *method*),
1

P

`PolygonAPIAccess` (*class in polygon_api_access*), 1

R

`reset_raw_data_tables()` (*poly-*
gon_api_access.PolygonAPIAccess *method*),
1

`reset_raw_data_tables2()` (*poly-*
gon_api_access.PolygonAPIAccess *method*),
2

S

`send_response()` (*poly-*
gon_api_access.PolygonAPIAccess *method*),
2

T

`ts_to_datetime()` (*poly-*
gon_api_access.PolygonAPIAccess *method*),
2